

Задача А. Максимальный поток

Имя входного файла: `maxflow.in`
Имя выходного файла: `maxflow.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Задан ориентированный граф, каждое ребро которого обладает целочисленной пропускной способностью. Найдите максимальный поток из вершины с номером 1 в вершину с номером n .

Формат входных данных

Первая строка входного файла содержит n и m — количество вершин и количество ребер графа ($2 \leq n \leq 100$, $1 \leq m \leq 1000$). Следующие m строк содержат по три числа: номера вершин, которые соединяет соответствующее ребро графа и его пропускную способность. Пропускные способности не превосходят 10^5 .

Формат выходных данных

В выходной файл выведите одно число — величину максимального потока из вершины с номером 1 в вершину с номером n .

Примеры

<code>maxflow.in</code>	<code>maxflow.out</code>
4 5 1 2 1 1 3 2 3 2 1 2 4 2 3 4 1	3

Задача В. Максимальный поток

Имя входного файла: flow2.in
Имя выходного файла: flow2.out
Ограничение по времени: 0.5 секунда
Ограничение по памяти: 256 мегабайт

Задан ориентированный граф, каждое ребро которого обладает целочисленной пропускной способностью. Найдите максимальный поток из вершины с номером 1 в вершину с номером n .

Формат входных данных

Первая строка входного файла содержит n и m — число вершин и ребер в графе ($2 \leq n \leq 500$, $1 \leq m \leq 10\,000$). Последующие строки описывают ребра. Каждое ребро задается тремя числами: начальная вершина ребра, конечная вершина ребра и пропускная способность ребра. Пропускные способности не превосходят 10^9 .

Формат выходных данных

Выведите величину максимального потока между вершинами 1 и n .

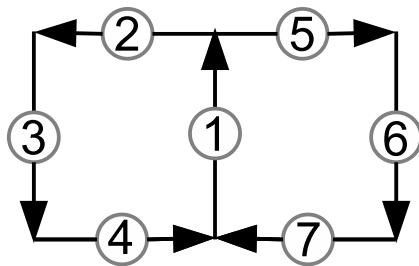
Примеры

flow2.in	flow2.out
4 5 1 2 1 1 3 2 3 2 1 2 4 2 3 4 1	3

Задача С. ЛКШ.Марафон

Имя входного файла: `running.in`
Имя выходного файла: `running.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В недождливый день в ЛКШ по плану должен пройти марафон. Он пройдёт по двум кругам: вокруг комповника и вокруг ГК. Условная и упрощенная схема дорожек в ЛКШ представлена на рисунке:



Про i -ю дорожку известна её максимальная пропускная способность r_i : если по ней будет бежать в среднем больше, чем r_i школьников в минуту, то будут происходить столкновения, и врачи прекратят марафон.

Но если по i -й дорожке будет бежать в среднем меньше, чем l_i школьников в минуту, то болельщикам, стоящим вдоль неё, будет скучно, и марафон прекратят культорги.

Может ли вообще пройти марафон? Если да, то какое максимальное число участников в минуту может бежать по центральной дорожке?

Формат входных данных

В первой строке содержатся два целых числа l_1 и r_1 , во второй — l_2 и r_2 , ..., в седьмой — l_7 и r_7 ($0 \leq l_i \leq r_i \leq 1000$).

Формат выходных данных

Если марафон провести невозможно, выведите «-1». Иначе выведите максимальное число школьников в минуту, которые могут бежать по центральной дорожке.

Примеры

running.in	running.out
50 100 10 30 20 40 0 100 0 100 20 40 0 20	50
0 0 0 30 0 40 0 100 0 100 0 40 0 20	0

Задача D. Нефтепроводы

Имя входного файла: oil.in
Имя выходного файла: oil.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Олигарх Вован, как и большинство других олигархов, занимается транспортировкой нефти из Западной Кукуляндии в Восточную Кукуляндию. В его владении находится огромная нефтедобывающая станция в Западной Кукуляндии, не меньшего размера нефтеперерабатывающая станция в Восточной Кукуляндии, а также система нефтепроводов, по которым нефть перегоняется из одной страны в другую. На столе у Вована лежит карта нефтепроводов. Хотелось бы знать, какое количество условных единиц нефти может транспортировать данная система. Каждый нефтепровод соединяет некоторую пару станций. На карте все станции пронумерованы, при этом добывающая станция имеет номер 1, перерабатывающая — номер N , а транзитные — номера от 2 до $N - 1$. Каждый нефтепровод может транспортировать ограниченное количество нефти, зато в любом направлении. Вован не знает, что Земля круглая, поэтому каждая станция на его карте имеет плоские координаты (x_i и y_i — координаты i -й станции). Нефтепроводы являются отрезками прямых. На карте пара нефтепроводов может пересекаться только по общей станции-вершине. Известно, что среди всех станций добывающая станция имеет наименьшую координату x , а перерабатывающая — наибольшую координату x .

Формат входных данных

В первой строке дано целое число N — количество станций на карте ($2 \leq N \leq 10000$). В следующих N строках перечислены координаты станций (x_i, y_i) через пробел. Координаты — целые числа, по модулю не превосходящие 10^8 . В следующей строке дано целое число M — количество нефтепроводов. Далее в M строках через пробел перечислены характеристики нефтепроводов — пара номеров станций, соединяемых нефтепроводом, а также пропускная способность нефтепровода в условных единицах — целое число от 1 до 10^8 . Гарантируется, что система нефтепроводов может транспортировать некоторое ненулевое количество нефти, но не может транспортировать более $2 \cdot 10^9$ условных единиц нефти.

Формат выходных данных

В первой строке выведите наибольшее количество условных единиц нефти, которое может транспортировать система Вована. В следующих M строках выведите план транспортировки — тройки чисел (A, B, C), означающие, что из станции A в станцию B должно течь C условных единиц нефти. Все нефтепроводы должны быть представлены в данном списке ровно один раз (даже те, по которым ничего не течёт). Число C во всех тройках должно быть неотрицательным.

Примеры

oil.in	oil.out
3	1
0 0	1 2 1
1 1	2 3 1
2 0	
2	
1 2 2	
2 3 1	

Задача Е. Пешки

Имя входного файла: `pawns.in`
Имя выходного файла: `pawns.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

В первом классе Глеб увлекался шахматами. К тому моменту он знал только лишь как ходит пешка: она может бить по диагонали влево-вверх и вправо-вверх, и ходить на клетку вверх только если та клетка не занята другой фигурой. О том, что пешка может превращаться в ферзя Глеб не подозревает. Поэтому он придумал свой вариант шахмат.

Игра идёт на доске с N строками и M столбцами ($1 \leq N \leq 100$, $1 \leq M \leq 100$) по следующим правилам. В нижней строке, имеющей номер 1, стоят P белых пешек, белых фигур на доске больше нет. На остальной части доски стоят разные чёрные фигуры (их названия Глеб не знает). Ходят только белые, их цель — побить все чёрные фигуры.

Как и в настоящих шахматах, если пешка Глеба бьёт чёрную фигуру, то она становится на её место, а побитая фигура убирается с доски. Считается, что Глеб выиграл, если он сумел побить белыми пешками все чёрные фигуры, в противном случае он проиграл. Помогите ему по заданной конфигурации всех фигур определить, сможет ли он выиграть, и, в случае успеха, выведите правильную последовательность ходов белых пешек.

Формат входных данных

Сначала вводятся четыре целых числа N , M , P , K ($1 \leq N \leq 100$, $1 \leq M \leq 100$, $0 \leq P \leq M$, $1 \leq K \leq 1000$, $K \leq (M - 1)N$). Далее записано P различных чисел — номера столбцов p_j ($1 \leq p_j \leq M$), в которых стоят белые пешки. Далее идут K различных пар целых чисел — координаты (строки и столбцы) чёрных фигур r_i , c_i ($2 \leq r_i \leq N$, $1 \leq c_i \leq M$).

Формат выходных данных

Если пешки не смогут съесть все фигуры, выведите единственное слово **NO**.

В противном случае в первую строку выведите **YES**, вторая строка должна содержать суммарное число перемещений C , последующие C строк — описание ходов пешек, по одному ходу на каждую строку. Каждый ход задаётся двумя координатами r , c пешки (номера строки и столбца), которая будет ходить, и символом m , принимающем три значения: **L**, **R**, **F** — побить вперед и влево, побить вперед и вправо, сделать шаг вперед соответственно. Данные о ходе следует выводить разделёнными одним пробелом, сначала координаты, потом тип хода.

Если последовательностей ходов несколько, выведите любой из них. Обратите внимание, что минимизировать количество перемещений не требуется.

Примеры

<code>pawns.in</code>	<code>pawns.out</code>
2 2 2 1 1 2 2 2	YES 1 1 1 R
3 3 2 2 1 3 3 1 3 3	NO

Задача F. Чокнутый профессор

Имя входного файла: `matan.in`
Имя выходного файла: `matan.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В Университете города М. проводят эксперимент. Преподаватели сами решают, что они будут читать в рамках того или иного курса. И вот преподаватель математического анализа (в простонародье — матана) оценил по некоторым критериям все известные ему темы в данном курсе. В результате этой ревизии каждой теме сопоставлено некоторое целое число (возможно, отрицательное) — полезность данной темы. Профессор хочет максимизировать суммарную полезность прочитанных им тем, но не все так просто. Для того что бы студенты поняли некоторые темы, необходимо, чтобы были прочитаны так же некоторые другие темы, так как некоторые доказательства базируются на фактах из других тем. Однако если существует цикл из зависимостей тем, то их все можно прочитать, и на качестве понимания материала студентами это не скажется.

Вас попросили составить список тем, которые профессор должен прочитать, таким образом, чтобы студенты все поняли, и суммарная полезность курса была максимальна.

Формат входных данных

Первая строка входного файла содержит одно число — N ($1 \leq N \leq 200$). Вторая строка содержит N целых чисел, не превосходящих по модулю 1000 — полезности каждой темы. Далее следуют N строк с описанием зависимостей тем. Каждое описание начинается количеством тем, которые необходимо понять для понимания данной темы. Потом следуют номера этих тем, разделенные пробелами.

Формат выходных данных

Выведите единственное число — максимально возможную суммарную полезность прочитанного материала.

Примеры

<code>matan.in</code>	<code>matan.out</code>
4 -1 1 -2 2 0 1 1 2 4 2 1 1	2
3 2 -1 -2 2 2 3 0 0	0